

Αλληλεπίδραση Ανθρώπου–Υπολογιστή

B10. Συστήματα διαλόγων φυσικής γλώσσας

(2023-24)

Ίων Ανδρουτσόπουλος

<http://www.aueb.gr/users/ion/>

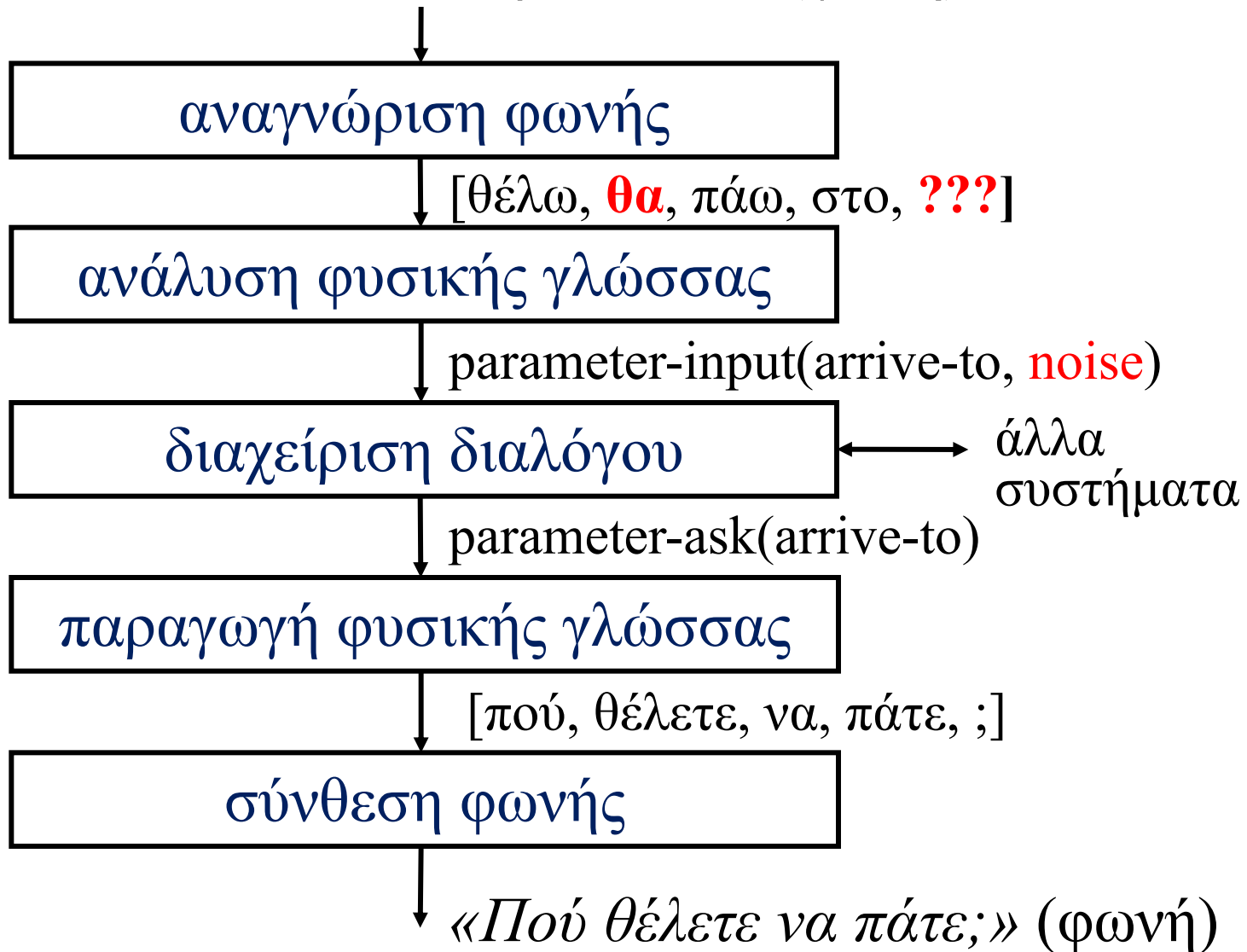
Οι διαφάνειες αυτές βασίζονται εν μέρει στην ύλη του βιβλίου *Speech and Language Processing* των D. Jurafsky και J.H. Martin, 2^η έκδοση (2009) και 3^η έκδοση (υπό προετοιμασία), Pearson Education.

Τι θα ακούσετε

- **Διαλογικά συστήματα** προφορικής και γραπτής φυσικής γλώσσας.
- **Συστήματα κουβέντας (chatbots)** ή για συγκεκριμένες δουλειές (**task oriented**).
- **Συστήματα διαλόγων** που χρησιμοποιούν **κανόνες, αυτόματα πεπερασμένων καταστάσεων, γραμματικές, πλαίσια, μηχανική μάθηση, προ-εκπαιδευμένα μεγάλα γλωσσικά μοντέλα.**

Συστήματα προφορικών διαλόγων

«Θέλω να πάω στο Ηράκλειο.» (φωνή)



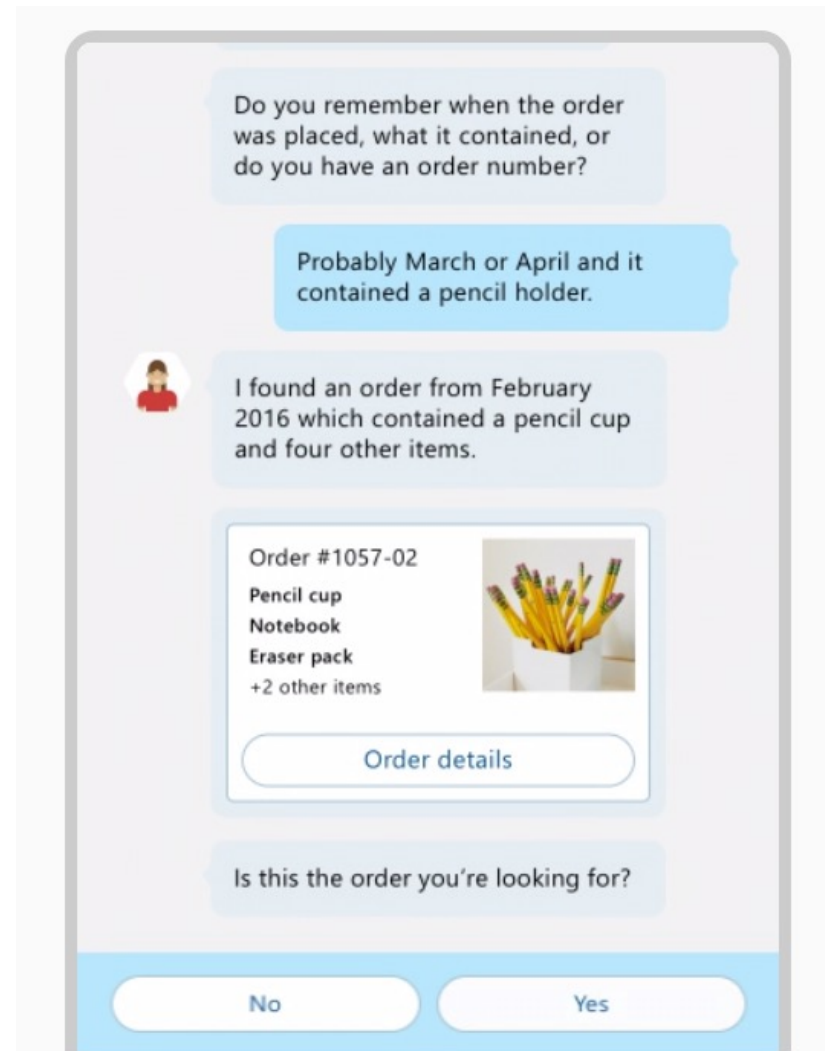
Συστήματα προφορικών διαλόγων

- **Πολλές εφαρμογές, όπως:**
 - κλείσιμο εισιτηρίων ή παροχή πληροφοριών,
 - πλοήγηση κατά την οδήγηση,
 - έλεγχος ή προγραμματισμός οικιακών συσκευών.
- Η προφορική γλώσσα διαφέρει πολύ από τη γραπτή.

*Ααα, καλημέρα. Θαα... Θα ήθελα να κλείσω για... εεεε...
Μπορείτε να μου πείτε εε... για Αθήνα... γύρω στις πέντε...
γι' αύριο μιλάω, έτσι; εεε... αν έχει θέσεις με Ολυμπιακή.*

Συστήματα γραπτών διαλόγων

- **Γραπτοί διάλογοι** (π.χ. σε Messenger, Viber).
- **Χωρίς τα λάθη της αναγνώρισης ομιλίας.**
- **Χωρίς ανάγκη εγκατάστασης ειδικών apps.**
- **Χωρίς να ακούν οι τριγύρω.**
- **Ευκολότερη ταυτοποίηση.**
- **Μικτές απαντήσεις κειμένου/κουμπιών και υπαλλήλων/συστήματος.**
- **Αλλά δυσκολίες όταν τα χέρια είναι απασχολημένα.**



Γενική μορφή διαλόγων

- **Εν γένει ένας συνομιλητής κάθε φορά έχει το λόγο.**
 - **Αποφεύγουν να μιλούν ταυτόχρονα** (turn taking).
 - Ενδέχεται ένας συνομιλητής να επιχειρήσει να **πάρει το λόγο** όσο μιλάει άλλος, αλλά υπάρχουν **σημεία του διαλόγου** όπου αυτό θεωρείται **πιο φυσιολογικό** (και ευγενικό).
 - Π.χ. όταν ο ομιλητής δείχνει να έχει **σχεδόν ολοκληρώσει** αυτό που λέει ή όταν **ρωτήσει** κάποιον άλλον.
- Συχνά αναμένεται **συγκεκριμένη δομή διαλόγου**.
 - Π.χ. χαιρετισμοί, αίτημα, συλλογή προτιμήσεων, επιβεβαίωση, αποχαιρετισμός.
- Σε **πολλά σημεία** αναμένεται ο συνομιλητής να **ανταποκριθεί με συγκεκριμένο τρόπο** (dialogic pairs).
 - Π.χ. χαιρετισμός–χαιρετισμός, αίτημα–αποδοχή ή απόρριψη.
 - Μια **καθυστερημένη ανταπόκριση** σε τέτοια σημεία μπορεί να εκληφθεί ως **απόρριψη, άρνηση ή αγένεια** (significant silence).

Γενική μορφή διαλόγων – συνέχεια

- Γενικά οι συνομιλητές χρειάζεται να **επιβεβαιώνουν** **διαρκώς** ότι τα όσα ακούν γίνονται **κατανοητά** (grounding).
 - Η παραμονή του συνομιλητή είναι **ασθενής επιβεβαίωση**.
 - **Ισχυρότερη επιβεβαίωση**: ο συνομιλητής **παίρνει το λόγο** **όπως αναμένεται**, **γνέφει καταφατικά**, **συμπληρώνει** ή **παραφράζει** τα λεγόμενα του συνομιλητή του κλπ.



Εικόνα από το CSLU Toolkit
(<http://cslu.cse.ogi.edu/toolkit/>).

Eliza (Weizenbaum, 1966)

Men are all alike.

IN WHAT WAY

They're always bugging us about something or other.

CAN YOU THINK OF A SPECIFIC EXAMPLE

Well, my boyfriend made me come here.

YOUR BOYFRIEND MADE YOU COME HERE

He says I'm depressed much of the time.

I AM SORRY TO HEAR THAT YOU ARE DEPRESSED

It's true that I am unhappy.

DO YOU THINK COMING HERE WILL HELP YOU NOT BE
UNHAPPY

(Πηγή: <http://web.mit.edu/STS001/www/Team7/eliza.html>)

- Το **πρώτο** και **από τα πιο γνωστά chatbots**.
 - Πολλές υλοποιήσεις και παραλλαγές ως σήμερα.
- Υπήρχαν χρήστες που **συζητούσαν μανιωδώς** μαζί του.

Προβλήματα με το Eliza

- Πολύ **επιφανειακή** επεξεργασία. Χρησιμοποιεί:
 - απλοϊκές **παραφράσεις** των λεγομένων του χρήστη,
 - απαντήσεις **πασπαρτού** (π.χ. «In what way?»),
 - **εντοπισμό** συγκεκριμένων λέξεων (π.χ. «my»).

HOW DO YOU DO. PLEASE STATE YOUR PROBLEM

I'm wondering whether computers can think.

WHY DO YOU MENTION COMPUTERS

Since you are one I thought I would ask you - can you think?

DID YOU THINK THEY MIGHT NOT BE ONE YOU

THOUGHT YOU WOULD ASK I -CAN I THINK

Let's talk just about you - not me. Can you think?

YOU LIKE TO THINK I - NOT YOU - DON'T YOU

Pardon?

(Πηγή: <http://web.mit.edu/STS001/www/Team7/eliza.html>)

Παραδείγματα κανόνων του Eliza

(από το βιβλίο των Jurafsky & Martin, 3^η έκδοση)

(0 you 0 me) → (WHAT MAKES YOU THINK I 3 YOU)

- Το 0 σημαίνει * (0 ή περισσότερες λέξεις).
- Το 3 αναφέρεται στις λέξεις της εισόδου που αντιστοιχούν στον 3^ο όρο του pattern, δηλαδή στις λέξεις του δεύτερου 0.

You hate me

WHAT MAKES YOU THINK I HATE YOU

(MEMORY my

(0 my 0 = LETS DISCUSS FURTHER YOUR 3)

(0 my 0 = EARLIER YOU SAID YOUR 3)

...)

- Αν ο χρήστης πει «my» (π.χ. «my father»), κράτα το τμήμα μετά το «my» ως πιθανό θέμα συζήτησης για τη συνέχεια...

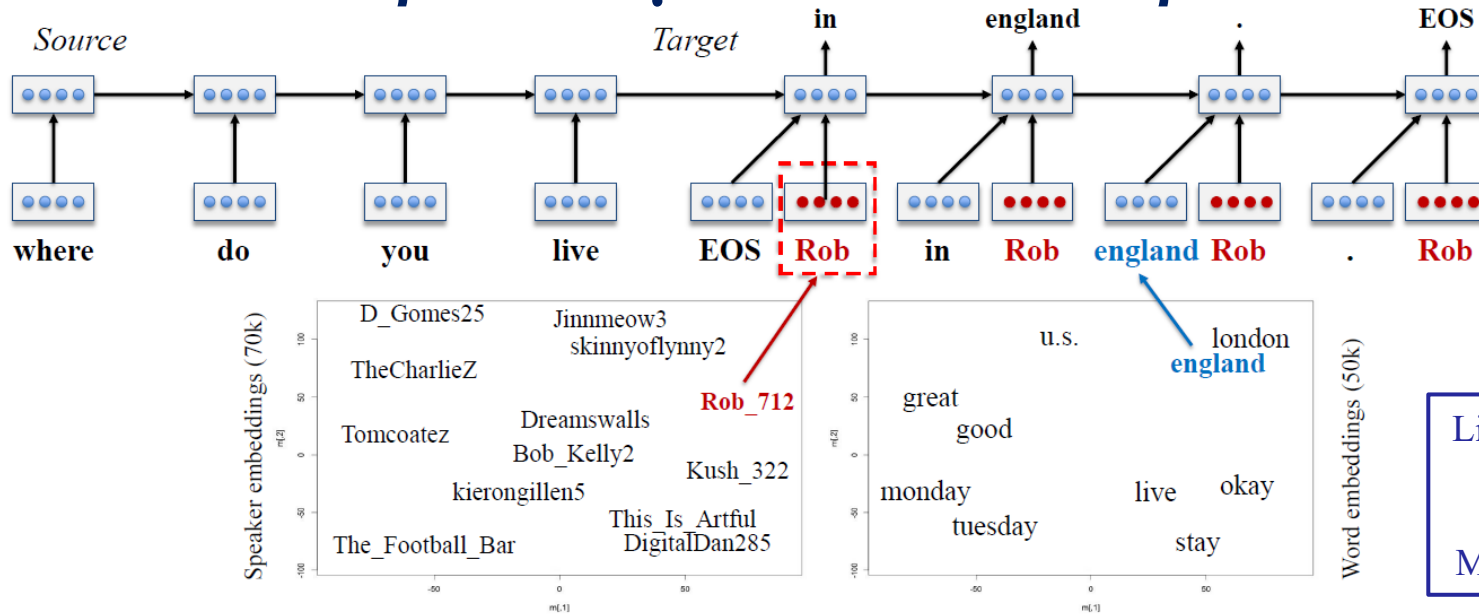
Chatbots βασισμένα σε ανάκτηση πληροφοριών

User:	Do you like Doctor Who?	
Cleverbot:	Yes, so funny.	
User:	What's your favorite show?	Από το βιβλίο των Jurafsky & Martin, 3 ^η έκδοση.
Cleverbot:	I don't have one.	
User:	What's your name?	
Cleverbot:	Jason I already told you.	Το σύστημα δεν θυμάται τι έχει πει προηγουμένως...
User:	No your real name.	
Cleverbot:	Stephanie IS my real name.	
User:	I thought it was Jason?	
Cleverbot:	How did you come to that conclusion.	
User:	Because you said so.	
Cleverbot:	I have to do my homework.	

Figure 29.4 Chatbots: A conversation between a user and Cleverbot, an IR-based chatbot (Carpenter, 2017).

- Χρησιμοποιούν **σώματα κειμένων διαλόγων**.
 - Π.χ. υπότιτλοι ταινιών, συζητήσεις σε κοινωνικά δίκτυα.
- Βρίσκουν την **απάντηση που μοιάζει περισσότερο με την πρόταση του χρήστη** (π.χ. edit distance, cosine similarity).
 - Ή την απάντηση που ακολουθεί (στο σώμα κειμένων) την πρόταση που μοιάζει περισσότερο με εκείνη του χρήστη.

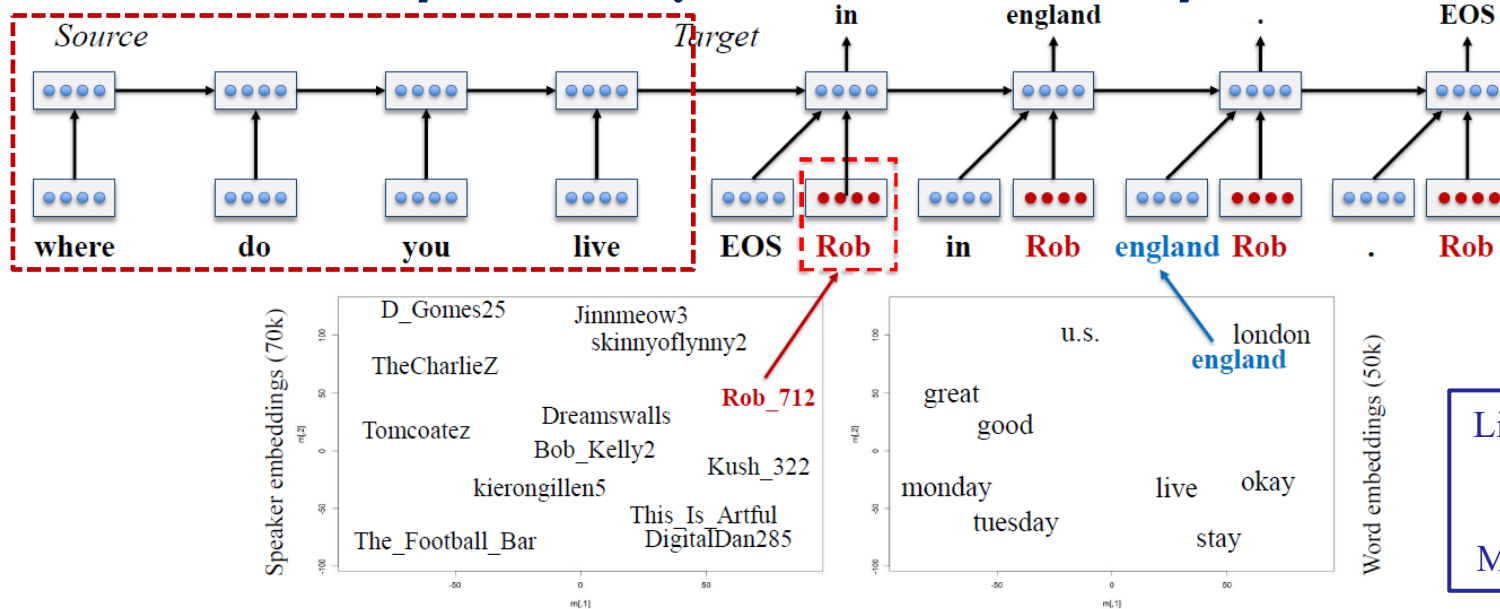
Chatbots βασισμένα σε νευρωνικά δίκτυα



Li et al., "A Persona-Based Neural Conversation Model", ACL, 2016.

- **Εκπαιδεύονται σε συλλογές διαλόγων (π.χ. υποτίτλων).**
 - Όπως στη μηχανική μετάφραση (βλ. ενότητες B6, B9) αλλά εδώ αντί για μετάφραση έχουμε παραγωγή της απόκρισης.
 - Εδώ με κωδικοποιητές/αποκωδικοποιητές RNN (ενότητα B6) αλλά μπορούν να χρησιμοποιηθούν και Transformers (ενότητα B9).
- Οι λέξεις και οι συνομιλητές παριστάνονται ως διανύσματα (word embeddings, speaker embeddings).
 - Τα διανύσματα των λέξεων ή/και των συνομιλητών μπορεί να τα μάθει το ίδιο το νευρωνικό δίκτυο κατά την εκπαίδευσή του.

Chatbots βασισμένα σε νευρωνικά δίκτυα



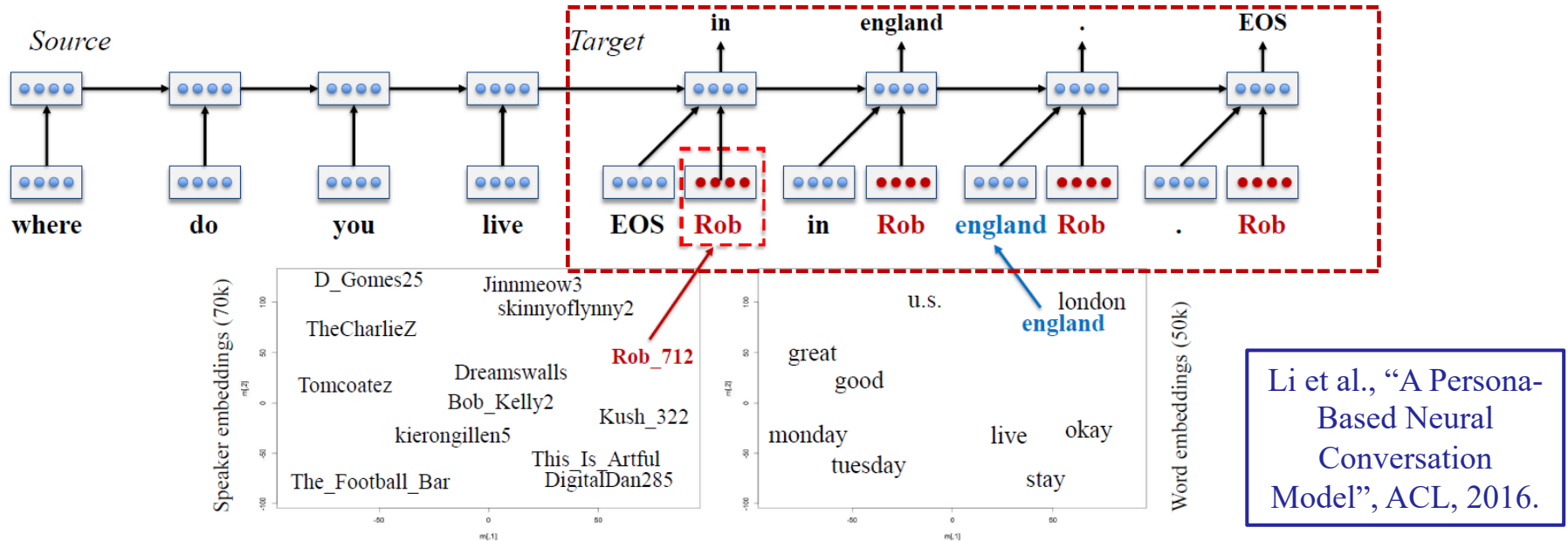
- **Κωδικοποιητής:** εδώ μετατρέπει την πρόταση του χρήστη σε ένα διάνυσμα που παριστάνει το νόημά της.
 - Σε κάθε βήμα (t) εξετάζει την προηγούμενη κατάσταση του RNN (\vec{h}_{t-1}) και την τρέχουσα λέξη (\vec{x}_t) της πρότασης του χρήστη και ενημερώνει την κατάσταση του RNN (\vec{h}_t):

$$\vec{h}_t = f(W_1\vec{h}_{t-1} + W_2\vec{x}_t + b_1)$$

Εναλλακτικά χρήση GRU ή LSTM (Ενότητα Β6).

- f συνάρτηση ενεργοποίησης, W_i πίνακες, b_j διανύσματα.
- Η τελική κατάσταση (εδώ \vec{h}_4) παριστάνει την πρόταση.

Chatbots βασισμένα σε νευρωνικά δίκτυα

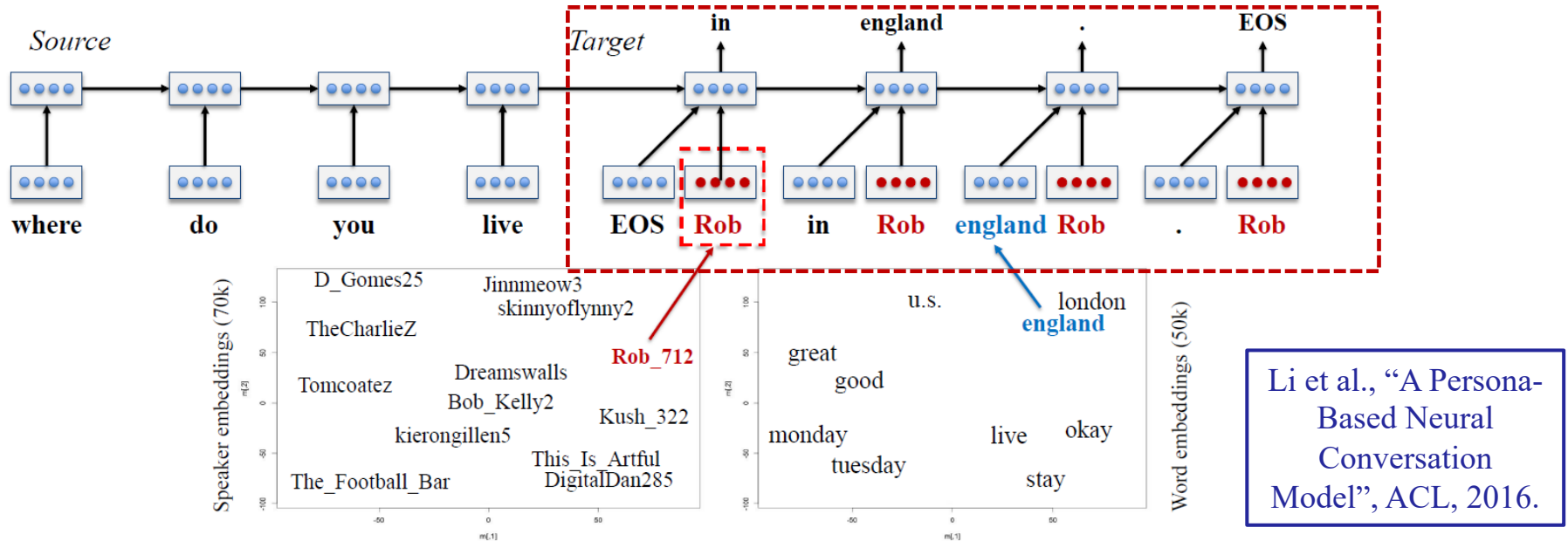


- **Αποκωδικοποιητής** (εδώ RNN και αυτός): Παράγει την **απόκριση του συστήματος**, λαμβάνοντας ως **αρχική κατάσταση** την **τελική κατάσταση του κωδικοποιητή**.
 - Σε κάθε βήμα, εξετάζει την προηγούμενη κατάσταση του αποκωδικοποιητή (\vec{g}_{t-1}), την προηγούμενη λέξη της απόκρισης (\vec{y}_{t-1}), το διάνυσμα του πράκτορα (\vec{a}) και παράγει τη νέα κατάσταση (\vec{g}_t) του αποκωδικοποιητή.

$$\vec{g}_t = f(W_3 \vec{g}_{t-1} + W_4 \vec{y}_{t-1} + W_5 \vec{a} + b_2)$$

Θα μπορούσαμε να προσθέσουμε και έναν μηχανισμό προσοχής (attention), όπως στη μηχανική μετάφραση της ενότητας Β6.

Chatbots βασισμένα σε νευρωνικά δίκτυα

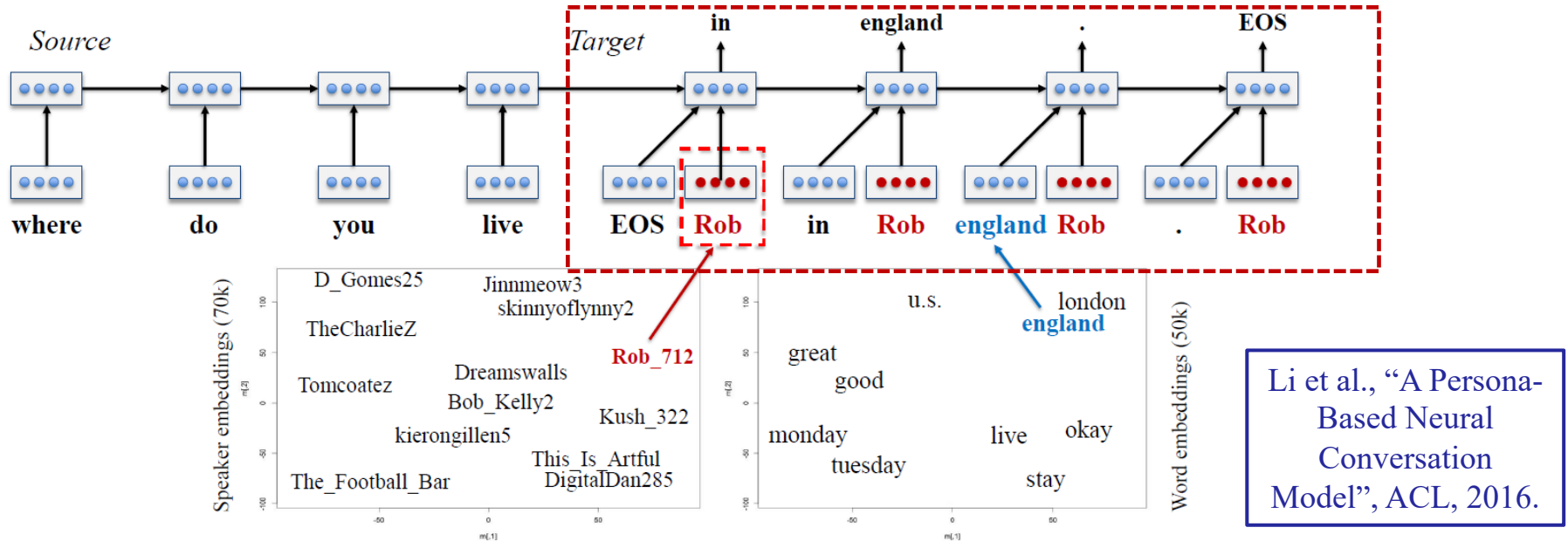


- **Αποκωδικοποιητής** (εδώ RNN και αυτός): Παράγει την **απόκριση του συστήματος**, λαμβάνοντας ως **αρχική κατάσταση** την **τελική κατάσταση του κωδικοποιητή**.
 - Σε κάθε βήμα, αφού δημιουργήσει τη **νέα κατάσταση** (\vec{g}_t), παράγει μια **κατανομή πιθανότητας** πάνω στο **λεξιλόγιο** V (αν $\vec{g}_t \in \mathbb{R}^d$, τότε $\vec{p}_t \in \mathbb{R}^{|V|}$, $W_6 \in \mathbb{R}^{|V| \times d}$, $b_3 \in \mathbb{R}^{|V|}$).

$$\vec{p}_t = \text{softmax}(W_6 \vec{g}_t + b_3)$$

- Η softmax φροντίζει να πάρουμε πιθανότητες (με άθροισμα 1).

Chatbots βασισμένα σε νευρωνικά δίκτυα



- **Αποκωδικοποιητής** (εδώ RNN και αυτός): Παράγει την **απόκριση του συστήματος**, λαμβάνοντας ως **αρχική κατάσταση** την **τελική κατάσταση του κωδικοποιητή**.
 - Στην πιο απλή περίπτωση, **σε κάθε βήμα παράγει λαίμαργα την πιθανότερη λέξη** (κατά την κατανομή \vec{p}_t) του **λεξιλογίου V** .

$$\vec{p}_t = \text{softmax}(W_6 \vec{g}_t + b_3)$$

- Στην πράξη χρησιμοποιούμε π.χ. **beam search** για να παραγάγουμε την **συνολικά πιθανότερη ακολουθία λέξεων**.

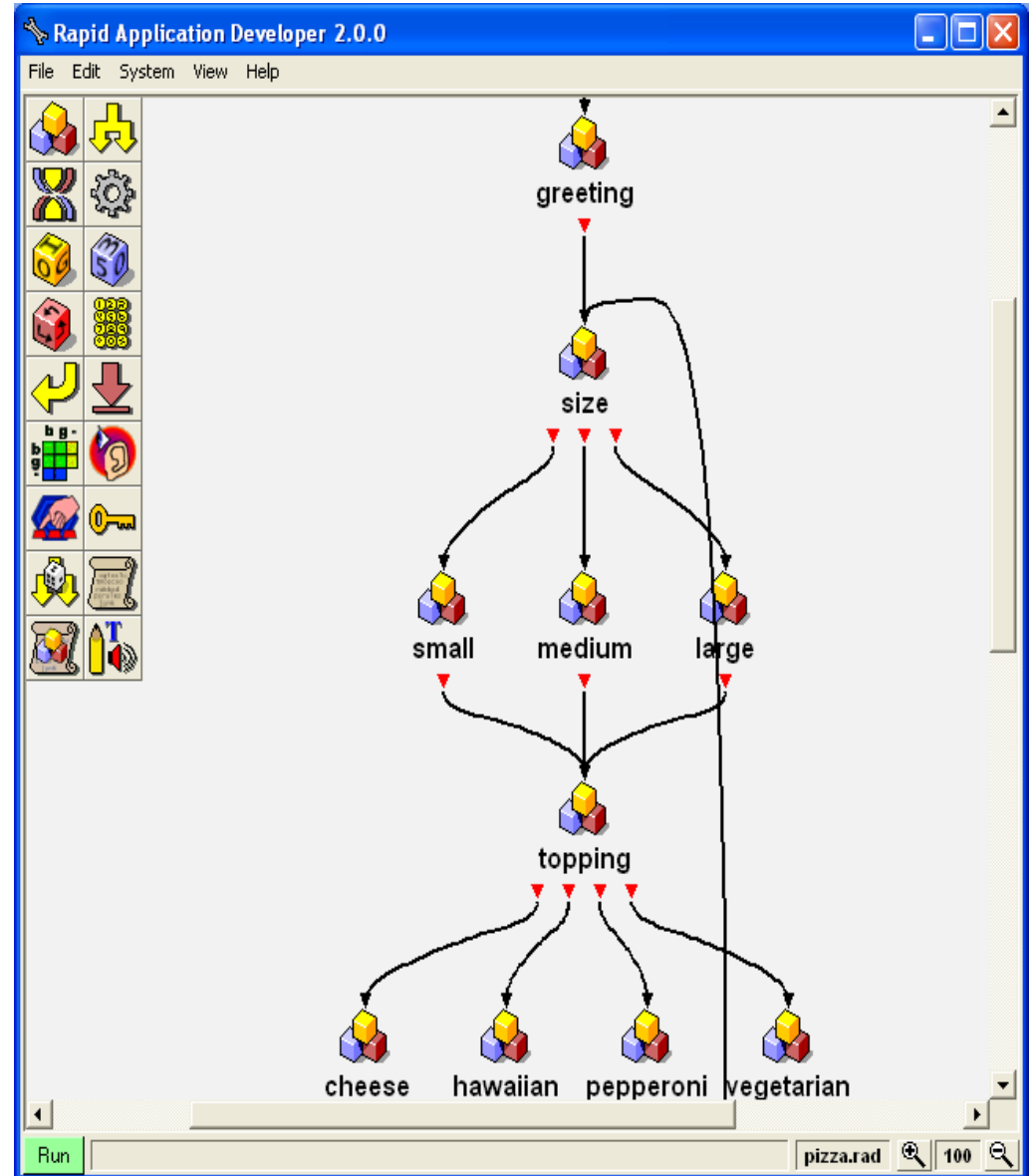
Διάλογοι για συγκεκριμένη δουλειά



Caption

Welcome to the C S L U pizza parlor!
Would you like a small, medium or
large pizza?
Please speak after the tone.

Εικόνες από το CSLU Toolkit
(<http://cslu.cse.ogi.edu/toolkit/>).



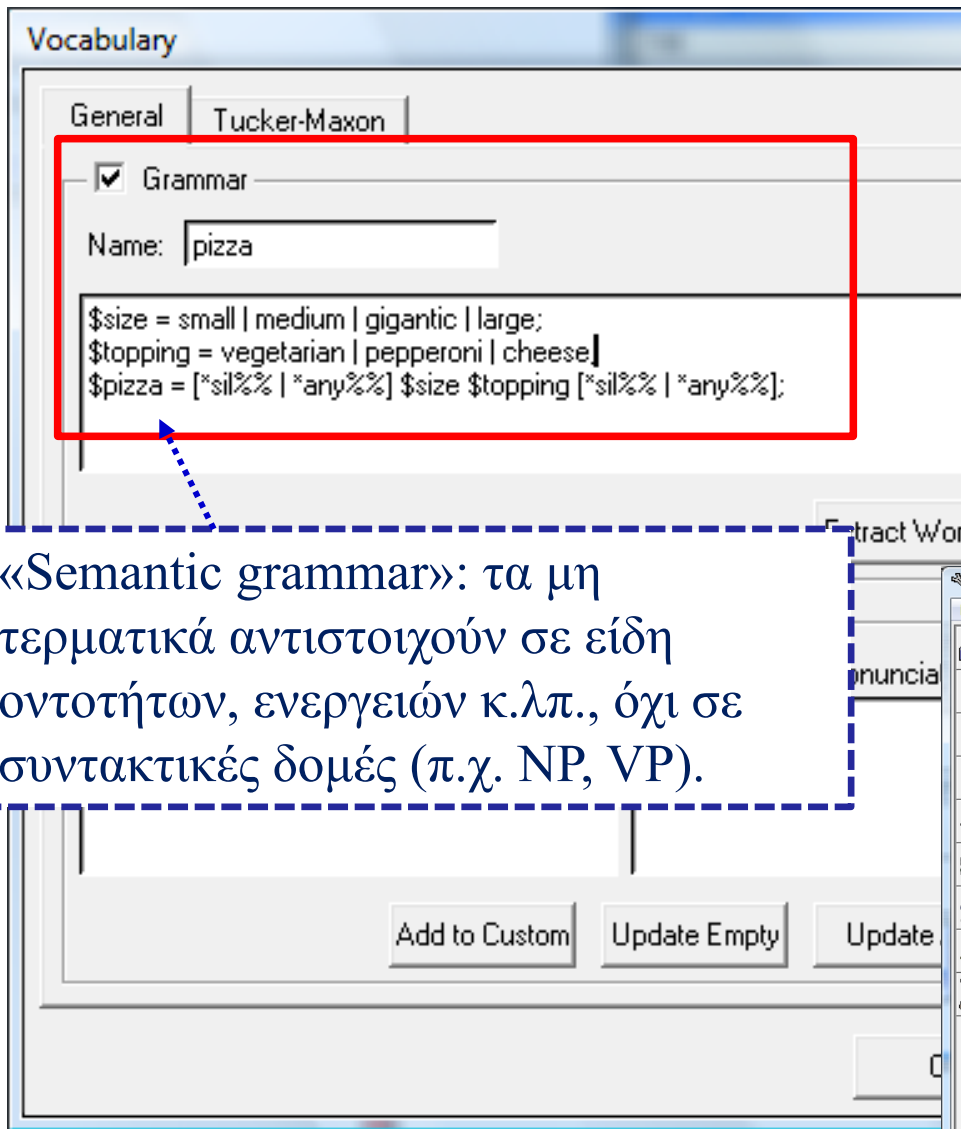
Διαχείριση διαλόγων με αυτόματα πεπερασμένων καταστάσεων

- **Απλά συστήματα** που προσπαθούν να βοηθήσουν τον χρήστη να εκτελέσει **συγκεκριμένη δουλειά** (task-based) συχνά χρησιμοποιούν **διαχειριστές διαλόγου** βασισμένους σε **αυτόματα πεπερασμένων καταστάσεων**.
 - Στην πιο **απλή περίπτωση**, σε **κάθε κατάσταση** το σύστημα **λέει κάτι** και **περιμένει συγκεκριμένες λέξεις**. Ανάλογα με τη λέξη που θα ακούσει, **πηγαίνει σε νέα κατάσταση**.
 - Σε κάποιες καταστάσεις εκτελούνται και **ενέργειες**, όπως π.χ. κλείσιμο εισιτηρίου.
- Για να μπορεί ο χρήστης να πει **ολόκληρες προτάσεις**, συχνά προστίθεται σε **κάθε κατάσταση** μια **γραμματική αναμενόμενων προτάσεων**.
 - **Υποβοηθούν την αναγνώριση φωνής** καθορίζοντας τις δυνατές ακολουθίες λέξεων του χρήστη σε **κάθε κατάσταση**.

Με διπλό κλικ στην κατάσταση *order*

The image shows two overlapping windows. The top window is titled "RAD prompt: order" and contains a "Prompt" field with the text "What size and type of pizza would you like?". Below the prompt field is a "-> Rec" button. The bottom window is titled "Rapid Application Developer 2.0" and displays a flowchart with four states: "start", "order", "verify", and "goodbye". The "start" state is connected to "order" by a dashed arrow, "order" to "verify" by a dashed arrow, and "verify" to "goodbye" by a solid arrow. A green arrow points down to "start", and a red arrow points down from "goodbye". A toolbar on the left of the RAD window contains various icons, including a yellow arrow pointing down, which is used to interact with the states in the flowchart. A "Run" button is visible at the bottom of the RAD window.

Με διπλό κλικ στην έξοδο της *order*



αρχικό σύμβολο: pizza

| : διάζευξη

[...]: προαιρετικό

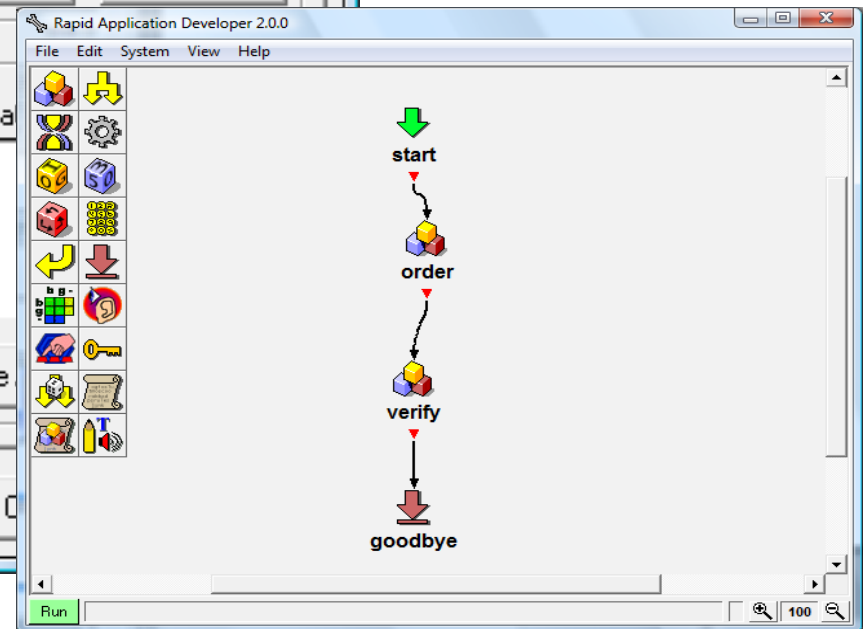
\$...: μη τερματικό σύμβολο

*sil: παύση

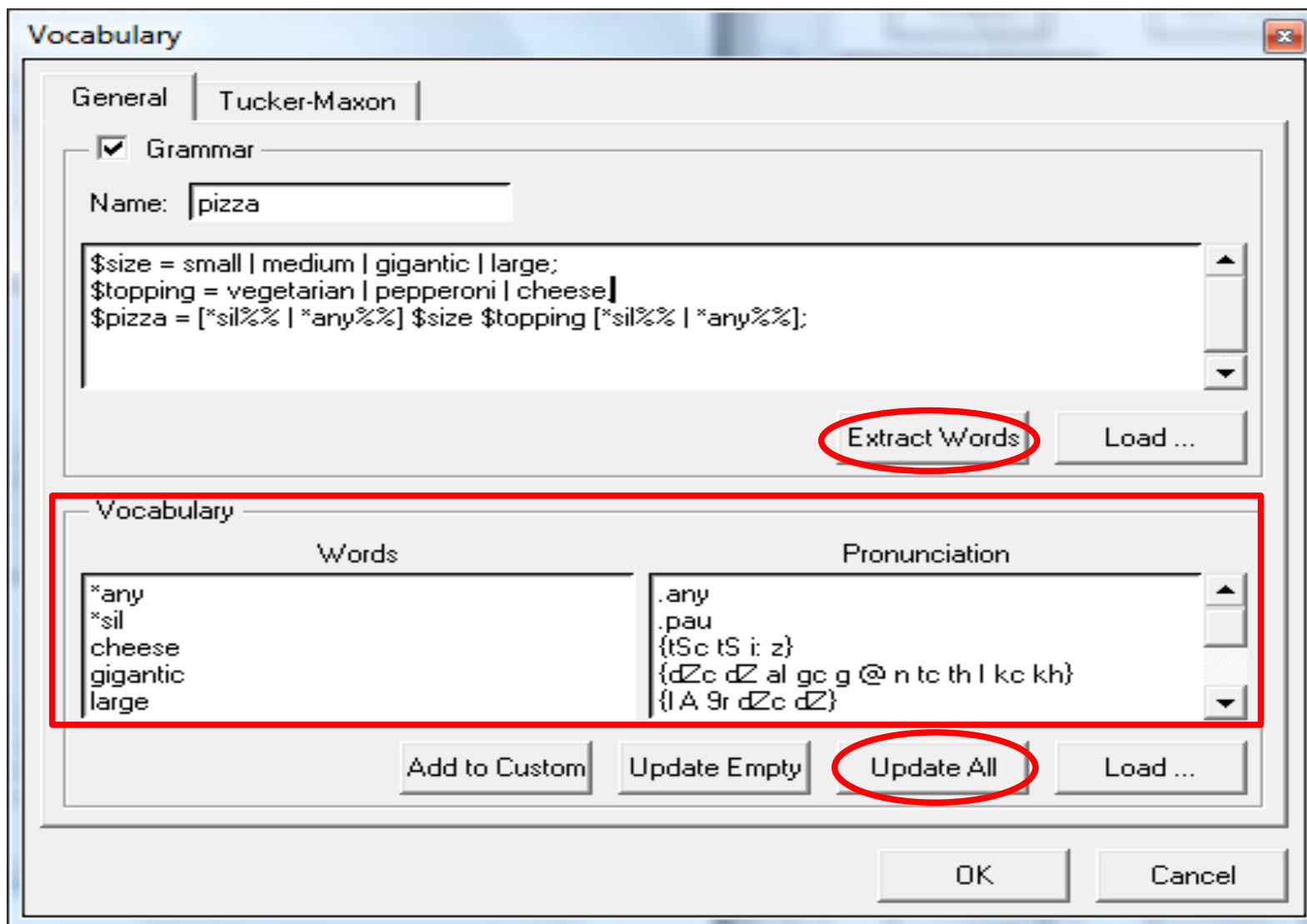
*any: λέξεις εκτός λεξικού

%%: μην περιλάβεις την αριστερή του λέξη στο αποτέλεσμα της αναγνώρισης

«Semantic grammar»: τα μη τερματικά αντιστοιχούν σε είδη οντοτήτων, ενεργειών κ.λπ., όχι σε συντακτικές δομές (π.χ. NP, VP).



Φωνητικό λεξικό



Διπλό κλικ στην κατάσταση *verify*

The image shows two overlapping windows from the CSLU Toolkit. The top window is titled "RAD prompt: verify" and contains a text area with the prompt "You ordered a \$order(recog) pizza." and a "Rec" button. The bottom window is titled "Rapid Application Developer 2.0.0" and displays a flowchart with nodes: start, order, verify, and goodbye. A red box highlights the "verify" node in the flowchart.

\$order(recog): Το αποτέλεσμα της αναγνώρισης φωνής στην κατάσταση order.

```
graph TD; start --> order; order --> verify; verify --> goodbye;
```

Εικόνες από το CSLU Toolkit
(<http://cslu.cse.ogi.edu/toolkit/>).

Παραγωγή αποκρίσεων

- Συχνά με χρήση απλών **σχεδιάτυπων** (templates).
 - Π.χ. «Ώστε θέλετε μια **\$order(recog)** πίτσα.»
- Εναλλακτικά με **παραγωγή κειμένων** (NL generation).
 - Π.χ. με **μεγάλο γλωσσικό μοντέλο** (LLM, βλ. παρακάτω).
 - Η **επιλογή περιεχομένου** (τι θα πούμε;) συνήθως γίνεται από το διαχειριστή διαλόγων.
- Ενδέχεται να προστίθενται και **ετικέτες** στο παραγόμενο κείμενο, οι οποίες να καθοδηγούν το **συνθέτη φωνής**.
 - Ετικέτες: π.χ. πες το με προσωδία **ερώτησης ή δήλωσης**, πες το με **ενθουσιασμό, θλίψη** κ.λπ., πες το ως **ημερομηνία, χρηματικό ποσό** κ.λπ., ως συγκεκριμένο **μέρος του λόγου** (π.χ. «objECT», αλλά «OBJect»), άφησε χρόνο μεταξύ **προτάσεων** κ.λπ.

Στρατηγικές επιβεβαίωσης

- **Ρητή επιβεβαίωση** (explicit confirmation).
 - «Είπατε ότι θέλετε να πάτε στην Αθήνα;»
 - «Ωστε θέλετε να πετάξετε προς την Αθήνα, αύριο, στις 9:00, με την Ολυμπιακή;»
 - Συνήθως προτιμότερη μόνο σε **κρίσιμα σημεία** ή όταν η **βεβαιότητα** αναγνώρισης/κατανόησης φωνής είναι **μικρή**.
- **Υπονοούμενη επιβεβαίωση** (implicit confirmation).
 - «Τι ώρα θέλετε να φύγετε για την Αθήνα;»
 - Προτιμότερη σε **μη κρίσιμα σημεία**, με **επαρκή βεβαιότητα**.
- **Απόρριψη** (rejection), αν η **βεβαιότητα** είναι **χαμηλή**.
 - Π.χ. αρχικά **σύντομη απόρριψη** («Συγγνώμη, τι είπατε;»), **κατόπιν πιο κατευθυνόμενη** («Συγγνώμη, πότε είπατε ότι θέλετε να φύγετε;»), με **τελικά εντελώς ελεγχόμενη** είσοδο (π.χ. «Παρακαλώ πληκτρολογήστε...»).

Πρωτοβουλία (initiative)

- Στα συστήματα με **αυτόματα** ως μοντέλα διαλόγου, ο διάλογος **καθοδηγείται κυρίως από το σύστημα**.
 - **Πρωτοβουλία συστήματος** (system initiative): το σύστημα έχει τον έλεγχο. Π.χ. το σύστημα προσφέρει συγκεκριμένες επιλογές και ο χρήστης επιλέγει ή το σύστημα ρωτά τις τιμές συγκεκριμένων παραμέτρων και ο χρήστης τις δίνει.
- Εναλλακτικά, ο διάλογος **καθοδηγείται από το χρήστη**.
 - **Πρωτοβουλία χρήστη** (user initiative): ο χρήστης έχει τον έλεγχο. Π.χ. θέτει ερωτήσεις ή δίνει εντολές και το σύστημα απαντά.
- Σε διαλόγους **μεταξύ ανθρώπων**, ο διάλογος συχνά **δεν είναι διαρκώς υπό τον έλεγχο του ίδιου συνομιλητή**.
 - **Μικτή πρωτοβουλία** (mixed initiative): π.χ. το σύστημα ξεκινά με «Πώς μπορώ να σας εξυπηρετήσω;», κατόπιν παίρνει την πρωτοβουλία όταν χρειάζεται («Τι ώρα θέλετε να πετάξετε;»).

Διαχείριση διαλόγων μέσω πλαισίων

- Η υποστήριξη **μικτής πρωτοβουλίας** είναι συχνά ευκολότερη με τη χρήση **πλαισίων**.

Πώς μπορώ να σας εξυπηρετήσω;

Θέλω να κλείσω μια πτήση για αύριο με την Ολυμπιακή προς την Αθήνα στις πέντε.

requestType(booking) \wedge date(tomorrow) \wedge carrier(oa) \wedge destination(ath) \wedge departTime(17:00)

Από πού θέλετε να αναχωρήσετε;

Π.χ. με
γραμματική
DCG.



requestType	booking	Θέλετε να κλείσετε ή να αλλάξετε εισιτήριο;
carrier	OA	Με ποια εταιρεία θέλετε να πετάξετε;
date	23/5/11	Ποια ημερομηνία θέλετε να πετάξετε;
departFrom		Από πού θέλετε να αναχωρήσετε;
destination	ATH	Ποιος είναι ο προορισμός σας;
departTime	17:00	Τι ώρα θέλετε να αναχωρήσετε;
...

Διαχείριση διαλόγων μέσω πλαισίων

- Το σύστημα χρησιμοποιεί μια **γραμματική** που καλύπτει προτάσεις για **οποιαδήποτε πεδία** του πλαισίου.
 - Ίσως και **πεδία** για τα οποία ο χρήστης **δεν είχε ρωτηθεί**.
 - Η γραμματική **εξάγει** και τις **τιμές** των **πεδίων**.
 - **Δυσκολότερη** έτσι όμως η **αναγνώριση φωνής**, γιατί η γραμματική επιτρέπει μεγαλύτερη ποικιλία προτάσεων.
- **Αν λείπει η τιμή ενός πεδίου**, το σύστημα παίρνει την **πρωτοβουλία** και **ρωτά για αυτό**.
 - **Για κάθε πεδίο**, το πλαίσιο παρέχει **κατάλληλη ερώτηση**.
- Μπορεί να χρησιμοποιούνται **πολλά πλαίσια**.
 - Ενδέχεται να χρησιμοποιείται **γράφος με κόμβους-πλαίσια** (π.χ. πρώτα πλαίσιο για κλείσιμο εισιτηρίων, μετά πλαίσιο για κλείσιμο ξενοδοχείου) ή/και **κανόνες ή ταξινομητές** που να **επιλέγουν το πλαίσιο** που θα ενεργοποιηθεί.

VoiceXML (<https://www.w3.org/Voice/>)

Προαιρετική
μελέτη

```
<noinput>
I'm sorry, I didn't hear you. <reprompt/>
</noinput>
```

Αν δεν ακούσαμε τίποτα, πες αυτό και επανάλαβε το προηγούμενο prompt.

```
<nomatch>
I'm sorry, I didn't understand that. <reprompt/>
</nomatch>
```

Αν δεν καταλάβαμε τι είπε ο χρήστης, πες αυτό και το προηγούμενο prompt.

```
<form>
  <block> Welcome to the air travel consultant. </block>
  <field name="origin">
    <prompt> Which city do you want to leave from? </prompt>
    <grammar type="application/x-nuance-gsl">
      [(san francisco) barcelona (new york)]
    </grammar>
    <filled>
```

Επιτρεπόμενες αποκρίσεις σε αυτό το σημείο. []=OR, ()=AND

```
      <prompt> OK, from <value expr="origin"/> </prompt>
    </filled>
  </field>
  <field name="destination">
    <prompt> And which city do you want to go to? </prompt>
    <grammar type="application/x-nuance-gsl">
      [(san francisco) barcelona (new york)]
    </grammar>
    <filled>
```

Πρωτοβουλία συστήματος. Ζητάμε σειριακά τις τιμές των τριών πεδίων.

```
      <prompt> OK, to <value expr="destination"/> </prompt>
    </filled>
  </field>
  <field name="departdate" type="date">
    <prompt> And what date do you want to leave? </prompt>
    <filled>
```

Επιβεβαίωση τιμής πεδίου.

```
      <prompt> OK, on <value expr="departdate"/> </prompt>
    </filled>
  </field>
  <block>
    <prompt> OK, I have you are departing from <value expr="origin"/>
      to <value expr="destination"/> on <value expr="departdate"/>
    </prompt>
    send the info to book a flight...
  </block>
</form>
```

Συνολική επιβεβαίωση.

Παράδειγμα από το βιβλίο των Jurafsky & Martin, 3^η έκδοση.

Μικτή πρωτοβουλία με VoiceXML

Προαιρετική
μελέτη

```
<noinput> I'm sorry, I didn't hear you. <reprompt/> </noinput>
<nomatch> I'm sorry, I didn't understand that. <reprompt/> </nomatch>
<form>
  <grammar type="application/x-nuance-gsl">
    <![CDATA[
      Flight ( ?[
        (i [wanna (want to)] [fly go])
        (i'd like to [fly go])
        ((i wanna)(i'd like a)] flight)
      ]
      [
        ( [from leaving departing] City:x) {<origin $x>}
        ( [(?going to)(arriving in)] City:x) {<destination $x>}
        ( [from leaving departing] City:x
          [(?going to)(arriving in)] City:y) {<origin $x> <destination $y>}
        ]
      ]?please
    )
    City [ [(san francisco) (s f o)] {return( "san francisco, california")}
          [(denver) (d e n)] {return( "denver, colorado")}
          [(seattle) (s t x)] {return( "seattle, washington")}
        ]
    ]]> </grammar>
  <initial name="init">
    <prompt> Welcome to the consultant. What are your travel plans? </prompt>
  </initial>
  <field name="origin">
    <prompt> Which city do you want to leave from? </prompt>
    <filled>
      <prompt> OK, from <value expr="origin"/> </prompt>
    </filled>
  </field>
  <field name="destination">
    <prompt> And which city do you want to go to? </prompt>
    <filled>
      <prompt> OK, to <value expr="destination"/> </prompt>
    </filled>
  </field>
  <block>
    <prompt> OK, I have you are departing from <value expr="origin"/>
      to <value expr="destination"/>. </prompt>
    send the info to book a flight...
  </block>
</form>
```

Γραμματική για όλο το πλαίσιο.

Flight → ...

City → ...

[]=OR, ()=AND, ?=προαιρετικό.

City:x = City με return(x).

<origin \$x> = γέμισμα πεδίου.

Παράδειγμα από το βιβλίο των
Jurafsky & Martin, 3^η έκδοση.

**Αν έχει μείνει κενό ένα πεδίο,
πες το αντίστοιχο prompt.**

Επιβεβαίωση πεδίου, αν έχει γεμίσει.

Συνολική επιβεβαίωση.

Αντί για γραμματικές...

- Αντί για γραμματικές, ενδέχεται να χρησιμοποιείται ένα **γλωσσικό μοντέλο** και **μέθοδοι αυτόματης επισημείωσης** ακολουθιών (token classification).
 - Το **γλωσσικό μοντέλο** βοηθά την **αναγνώριση φωνής** να δώσει **προτεραιότητα** σε **πιο πιθανές προτάσεις**.
 - Η **αυτόματη επισημείωση** σημειώνει **φράσεις** που αντιστοιχούν σε **τιμές πεδίων** ενός πλαισίου. Π.χ. με RNNs, CNNs, Transformers, όπως κατά τον εντοπισμό ονομάτων οντοτήτων (βλ. ενότητες B6, B8, B9).
 - Απαιτείται και **κανονικοποίηση** των φράσεων-τιμών (π.χ. με απλές γραμματικές, βλ. 6^η άσκηση μελέτης ενότητας B3).

[ignore Καλημέρα θα ήθελα] [noise #\$\$\$@#\$\$] [ignore παρακαλώ να]
[requestType κλείσω] [noise @#\$\$\$] [destination για Αθήνα] [noise @#\$\$@]
[departTime στις πέντε] [noise #\$\$\$@#\$\$] [date αύριο] [noise \$@###\$\$]
[carrier με Ολυμπιακή]

Γνωστική περιοχή και πρόθεση

- Σε συστήματα όπως τα Siri, Alexa, Cortana, Google Now χρειάζεται να καταλάβουμε τη **γνωστική περιοχή (domain)** και την **πρόθεση (intent)** του χρήστη και να συμπληρώσουμε τα **πεδία** του αντίστοιχου πλαισίου.

Show me morning flights from
Boston to San Francisco on Tuesday

DOMAIN: AIR-TRAVEL
INTENT: SHOW-FLIGHTS
ORIGIN-CITY: Boston
ORIGIN-DATE: Tuesday
ORIGIN-TIME: morning
DEST-CITY: San Francisco

Παραδείγματα
από το βιβλίο των
J&M, 3^η έκδοση.

Wake me tomorrow at 6

DOMAIN: ALARM-CLOCK
INTENT: SET-ALARM
TIME: 2017-07-01 0600-0800

- Μπορεί να γίνονται όλα μέσω **γραμματικών** ή να υπάρχουν **ξεχωριστοί ταξινομητές** που να μαντεύουν τη **γνωστική περιοχή, την πρόθεση, τα πεδία**.

Ελεύθερα διαθέσιμο: wit.ai

wit.ai

+ Apps Docs Help

ionandr / MyFirstApp / DONE

Understanding

Samples

Inbox

Logs

Settings

Filter by: All Entities All samples

“ Search through your samples.

Text

I would like a **spinach** pizza

intent pizza_type

Add a new entity

Παράδειγμα εκαίδευσης.

Σωστή πρόθεση (intent).

Τύπος και τιμή πεδίου (slot)

pizza_order

spinach

“ I want a pepperoni pizza ×

“ I would like a margherita pizza ×

“ I would like a pepperoni pizza please ×

“ I would like a cheese pizza ×

“ What is the weather in London? ×

“ What is the weather in Athens? ×

“ Set the temperature to 50 degrees ×

Δείτε και την ιστοσελίδα συνδέσμων του μαθήματος για παρόμοια εργαλεία (π.χ. Rasa).

Εύρεση γνωστικής περιοχής μέσω LLM

Prompt	<p>Determine which domain is considered in the following dialogue situation. Choose exactly one domain from this list: restaurant, hotel, attraction, taxi, train Answer with only one word, the selected domain from the list. You have to always select the most probable domain.</p> <p>—— Example 1: —— Customer: I need a cheap place to eat Assistant: We have several not expensive places available. What food are you interested in? Customer: Chinese food. Domain: restaurant</p> <p>—— Example 2: —— Customer: What is the address? Assistant: It's 123 Northfolk Road. Customer: That's all. I also need a train from London. Domain: train</p> <p>Now complete the following example: Customer: I am looking for a cheap place to stay. Domain:</p>	Οδηγίες (instructions)
Output:	hotel	Παραδείγματα (demonstrators, in-context few-shot examples)
		Είσοδος για την οποία ζητείται απόκριση και η απόκριση του LLM

Table 4: A prompt used for domain detection for MultiWOZ. It contains **task definition**, **domains description**, **static examples** and **user utterance**.

Παράδειγμα από την εργασία των V. Hudeček και O. Dusek, «Are Large Language Models All You Need for Task-Oriented Dialogue?», SIGDIAL 2023 (<https://aclanthology.org/2023.sigdial-1.21/>).

Εύρεση τιμών πεδίων μέσω LLM

Prompt	<p>Definition: Capture entity values from last utterance of the conversation according to examples. Capture pair "entity:value" separated by colon and no spaces in between. Separate entity:value pairs by hyphens. If not specified, leave the value empty. Values that should be captured are:</p> <ul style="list-style-type: none">- "pricerange": the price of the hotel- "area" that specifies the area where the hotel is located (north/east/west/south/centre)- "internet" that specifies if the hotel has internet (yes/no)- "parking" that specifies if the hotel has parking (yes/no)- "stars" that specifies the number of stars the hotel has (1/2/3/4/5)- "type" that specifies the type of the hotel (hotel/bed and breakfast/guest house) <p>[history] Customer: "I want a cheap place to stay."</p>
Output:	pricerange:"cheap"

Οδηγίες για συγκεκριμένη γνωστική περιοχή (domain)

Table 5: A zero-shot version of the prompt used for state update prediction for MultiWOZ 2.2. It contains **task definition**, **domain description**, **dialogue history** and **user utterance**.

Στη θέση του [history] θα έμπαινε όλος ο προηγούμενος διάλογος.

Είσοδος για την οποία ζητείται απόκριση και η απόκριση του LLM

Θα μπορούσαμε να είχαμε προσθέσει πάλι μερικά παραδείγματα (**few-shot examples**) στο prompt.

Παράδειγμα από την εργασία των V. Hudeček και O. Dusek, «Are Large Language Models All You Need for Task-Oriented Dialogue?», SIGDIAL 2023 (<https://aclanthology.org/2023.sigdial-1.21/>).

End-to-end dialogue systems based on Transformers

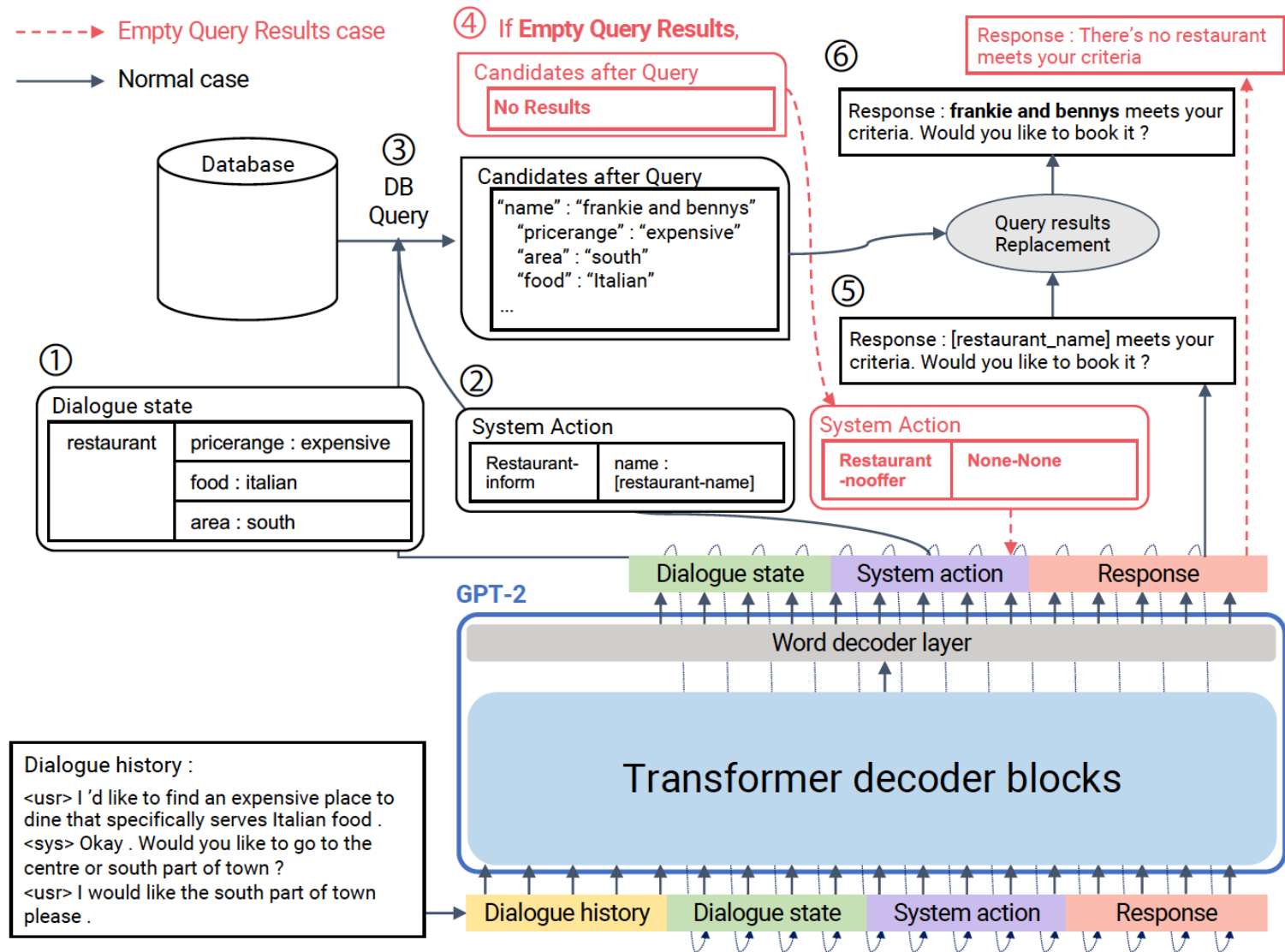
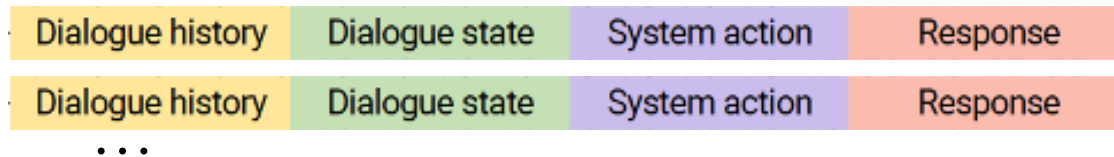


Figure from Ham et al. (2020), "End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2" (ACL 2020, <https://aclanthology.org/2020.acl-main.54/>).

Dialogue systems based on Transformers

- A **pre-trained Transformer** is **fine-tuned** on examples containing:
 - The **dialogue history** (the previous user and system utterances).
 - The correct **current dialogue state** (frame, serialized) given the history.
 - The correct **system action** (e.g., a database query using values from the dialogue state as search constraints).
 - The **response** to the user, as a **template** where values retrieved from the database (e.g., [restaurant_name]) will be replaced during post-processing.



- The system is **fine-tuned** to **generate** (auto-complete) the **dialogue state**, **system action**, **response**, given the dialogue history.
 - The paper of the previous slide used a **decoder-only GPT-2**, **pre-trained as a language model** (section B9). Encoder-decoder models could be used too.
 - With a model like **ChatGPT**, one could also use **prompting with few-shot examples** instead or in addition to fine-tuning.

Hugging Face agents

Εκτός
εξεταστέας ύλης

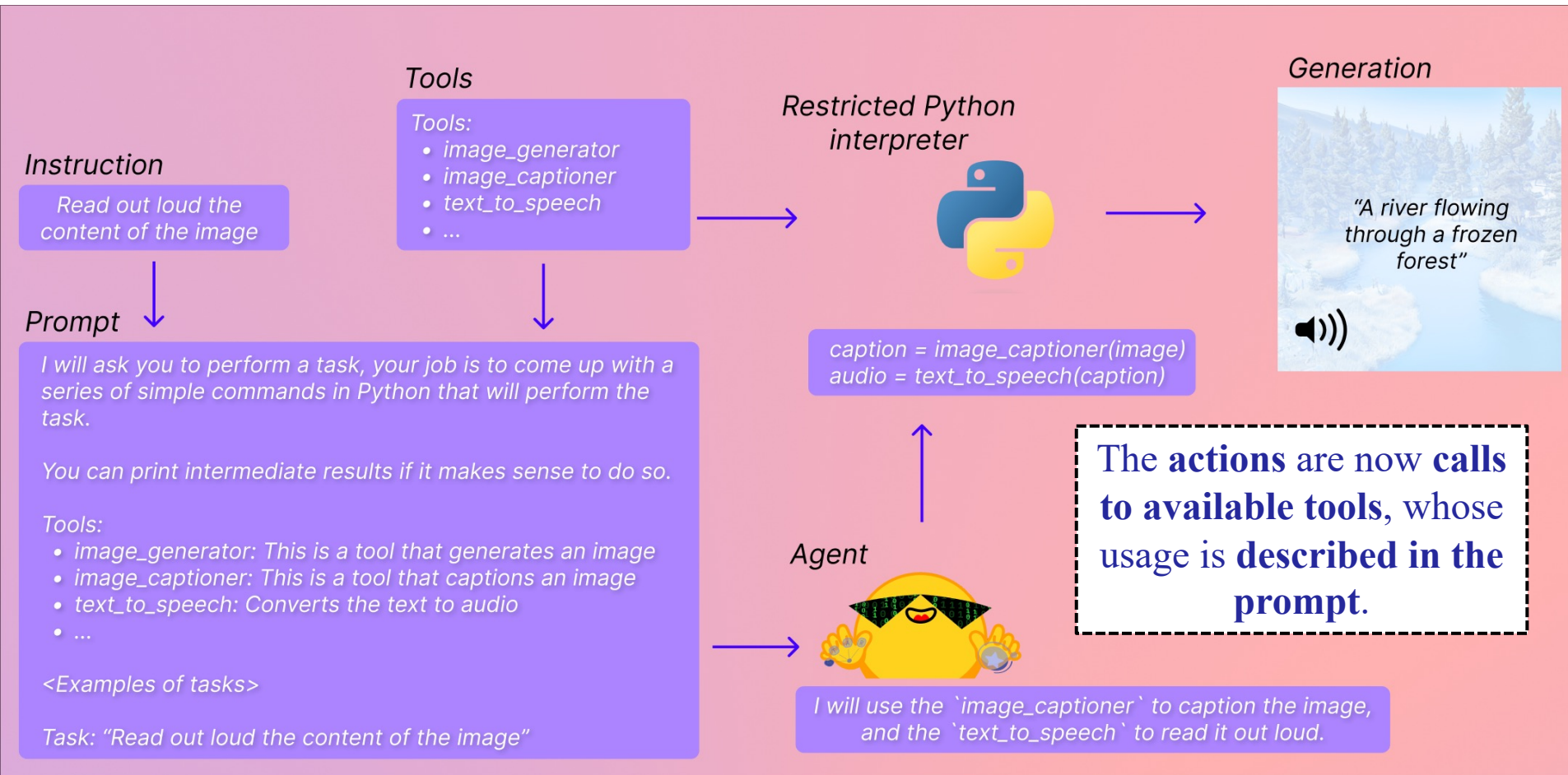


Figure from https://huggingface.co/docs/transformers/transformers_agents.

Hugging Face agents

Εκτός
εξεταστέας ύλης

```
audio = agent.run("Read out loud the summary of http://hf.co")  
play_audio(audio)
```

==Explanation from the agent==

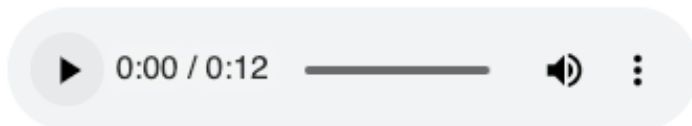
I will use the following tools: `text_downloader` to download the text from the website, `summarizer` to create a summary of the text, and `text_reader` to read it out loud.

==Code generated by the agent==

```
text = text_downloader("https://hf.co")  
summarized_text = summarizer(text)  
print(f"Summary: {summarized_text}")  
audio_summary = text_reader(summarized_text)
```

==Result==

Summary: Hugging Face is an AI community building the future. More than 5,000 organizations are using Hugging Face's AI chat models. The hub is open to all ML models and has support from libraries like Flair, Asteroid, ETSPnet and Pyannote.



Example from https://huggingface.co/docs/transformers/transformers_agents.

Διάβασμα

- Μεγάλο μέρος της ύλης αυτής της ενότητας καλύπτεται από το κεφ. 24 της 3^{ης} έκδοσης (υπό προετοιμασία) του βιβλίου «Speech and Language Processing» των Jurafsky & Martin, Pearson Education.
 - <http://web.stanford.edu/~jurafsky/slp3/>
 - Για τις εξετάσεις χρειάζεται να γνωρίζετε μόνο όσα αναφέρουν οι διαφάνειες του μαθήματος.
 - Μπορείτε να συμβουλευτείτε (στη βιβλιοθήκη του ΟΠΑ) και το κεφ. 24 της 2^{ης} έκδοσης, ενότητες 24.1–24.4.

